

All Mouse Cursors

```
using System;
using System.Drawing;
using System.Windows.Forms;

class MouseCursors: Form
{
    Cursor[] acursor =
    {
        Cursors.AppStarting, Cursors.Arrow, Cursors.Cross,
        Cursors.Default, Cursors.Hand, Cursors.Help,
        Cursors.HSplit, Cursors.IBeam, Cursors.No,
        Cursors.NoMove2D, Cursors.NoMoveHoriz, Cursors.NoMoveVert,
        Cursors.PanEast, Cursors.PanNE, Cursors.PanNorth,
        Cursors.PanNW, Cursors.PanSE, Cursors.PanSouth,
        Cursors.PanSW, Cursors.PanWest, Cursors.SizeAll,
        Cursors.SizeNESW, Cursors.SizeNS, Cursors.SizeNWSE,
        Cursors.SizeWE, Cursors.UpArrow, Cursors.VSplit,
        Cursors.WaitCursor
    };
    string[] astrCursor =
    {
        "AppStarting", "Arrow", "Cross",
        "Default", "Hand", "Help",
        "HSplit", "IBeam", "No",
        "NoMove2D", "NoMoveHoriz", "NoMoveVert",
        "PanEast", "PanNE", "PanNorth",
        "PanNW", "PanSE", "PanSouth",
        "PanSW", "PanWest", "SizeAll",
        "SizeNESW", "SizeNS", "SizeNWSE",
        "SizeWE", "UpArrow", "VSplit",
        "WaitCursor"
    };
};

public static void Main()
```

```

{
Application.Run(new MouseCursors());
}
public MouseCursors()
{
Text = "Mouse Cursors";
ResizeRedraw = true;
}
protected override void OnMouseMove(MouseEventArgs mea)
{
int x = Math.Max(0, Math.Min(3, mea.X / (ClientSize.Width /
4)));
int y = Math.Max(0, Math.Min(6, mea.Y / (ClientSize.Height /
7)));

Cursor.Current = acursor[4 * y + x];
}
protected override void OnPaint(PaintEventArgs pea)
{
Graphics grfx = pea.Graphics;
Brush brush = new SolidBrush(ForeColor);
Pen pen = new Pen(ForeColor);
StringFormat strfmt = new StringFormat();

strfmt.LineAlignment      =      strfmt.Alignment      =
StringAlignment.Center;

for (int y = 0; y < 7; y++){ for (int x = 0; x < 4; x++) {
Rectangle rect = Rectangle.FromLTRB( x * ClientSize.Width / 4,
y * ClientSize.Height / 7, (x + 1) * ClientSize.Width / 4, (y
+ 1) * ClientSize.Height / 7); grfx.DrawRectangle(pen, rect);
grfx.DrawString(acursor[4 * y + x], Font, brush, rect,
strfmt); } } } } [/csharp]

```